CS551 Advanced Software Engineering

PROJECT INCREMENT-3

Project Name: PASEO

Group Details: Team-2

Amulya Pindi

Naresh Pogakula

Vinutha Muthyala

Megha Sai Reddy Bodimani

I INTRODUCTION:

Carpooling is sharing of car journeys such that more than people travel together in a car towards the same destination. Carpooling has proved to be a major triumph in reducing the emissions of greenhouse gases as it reduces the number of cars on road, further it has also proved to be a blessing to those who do not own a private car.

The application "PASEO" is a carpooling android application with security measures to ensure that the user feels secured all times. We register the rider and the passenger and validate the details of the users by checking their passport number, Driver's license, car details etc. The rider gives the timings of his ride and the passenger picks up the ride based on his convenience. We further ensure full security to the users by providing unique features like sharing current location, tracking the ride in case of emergency providing them a button to give alerts based on location and time.

II OBJECTIVES

There are a lot of impediments faced by people who travel to different parts of the city in their own transport which kind of makes them irksome and there are also people who do not have their own transport and pay a bomb to commute in the city. The amount of environmental pollution that is caused by single driver cars is phenomenal and is on the rise. On an average American spends 40 hours each year stuck in traffic which further adds to our woes.

According to a survey by carpooling just twice a week about 1,600 pounds of greenhouse gases can be kept out of the air each year. Further carpooling cuts down the cost of both the rider and the passenger, makes their life and also for the generations to come.

FEATURES:

- REGISTER AND LOGIN

User can register themselves as a rider or a passenger through the application by giving their details.

- VALIDATION

The application validates certain details of the rider by checking his history, checks if he has a valid license and checks certain details of the car.

- ANONYMITY

The application ensures possible anonymity of both the rider and the passenger as only their names will be revealed. They can contact each other through the app to ensure certain safety.

- RIDE TRACKING

The user can track his ride through the app and can also share his location on the click of a single button to his desired friends who can also track the users ride to ensure to safety.

- EMERGENCY ALERT

The user can just pop out an emergency signal to the police and his family in case of any accident or in danger on the click of a button or on the utterance of a certain keyword.

- SAFETY METER

The user can see how safe an area is during different times of the day in which they are travelling, so they can be alert at each and every point. The application pops out a warning or alert on case if the area is highly unsafe.

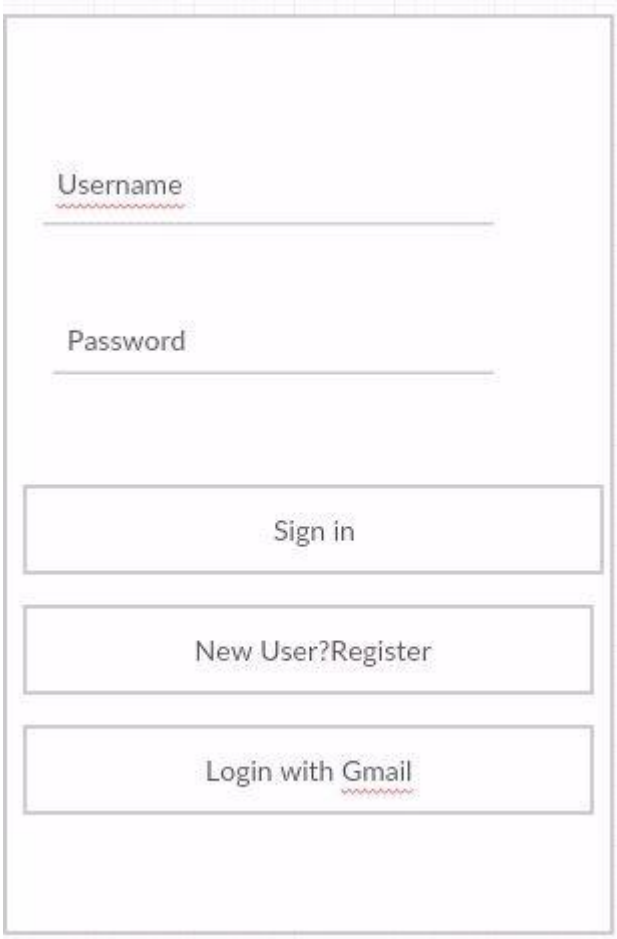
EXISTING API's:

1. Google Maps API
2. Mongo DB API

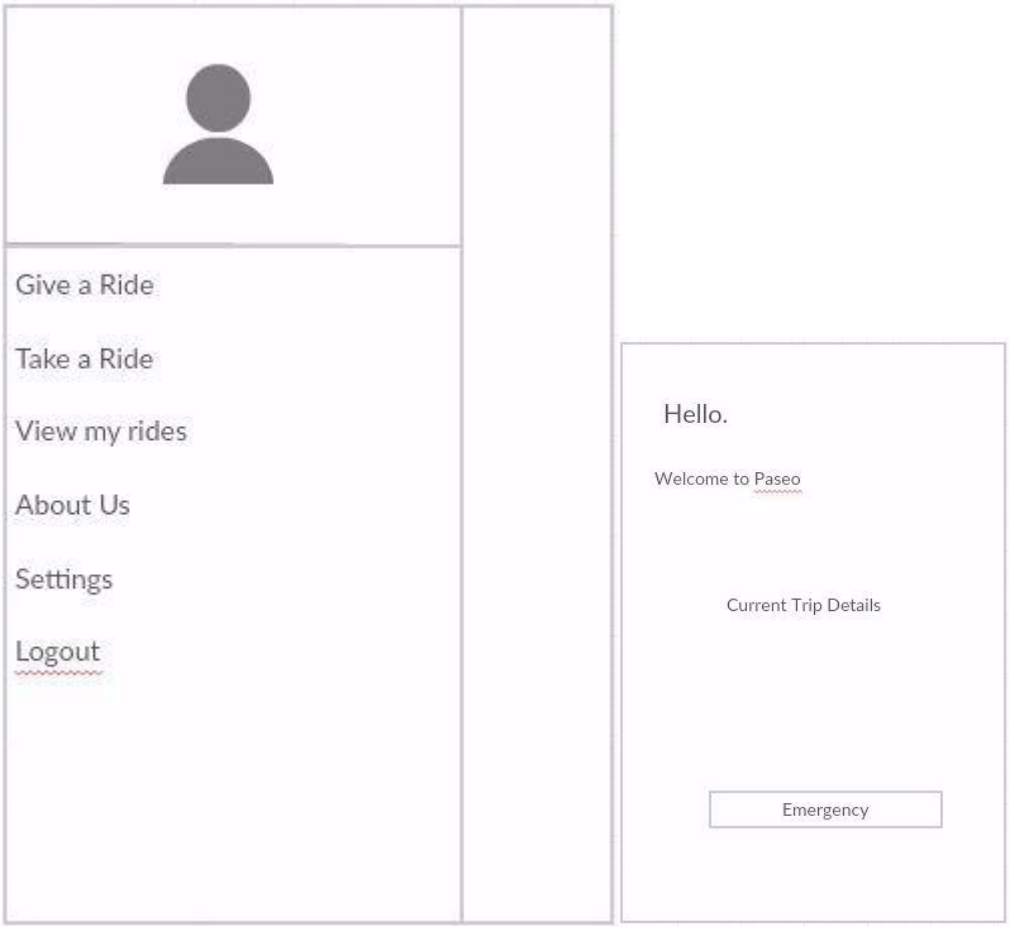
DETAIL DESIGN OF FEATURES:

WIREFRAMES AND MOCKUPS

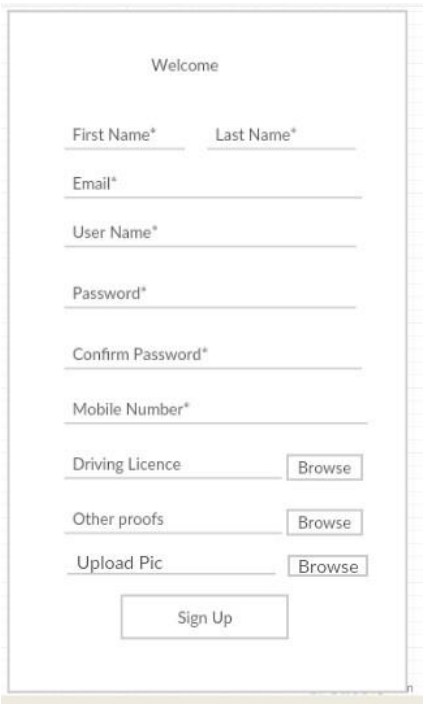
1. The login Page is used by the users to login the application and New-Users can click register to create an account in the application. Users can also login with either Gmail or Facebook. A link About Us is provided to give user information about the app.



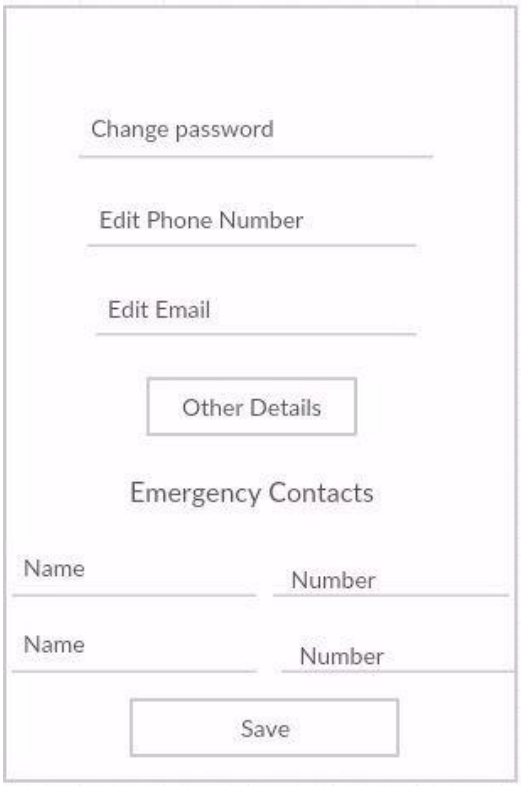
2. Registered user would be navigated to the Home screen as below where the user can perform multiple functions.



3. The Registration page is used by the New-Users to enter their details and save them to data base for creating the account.

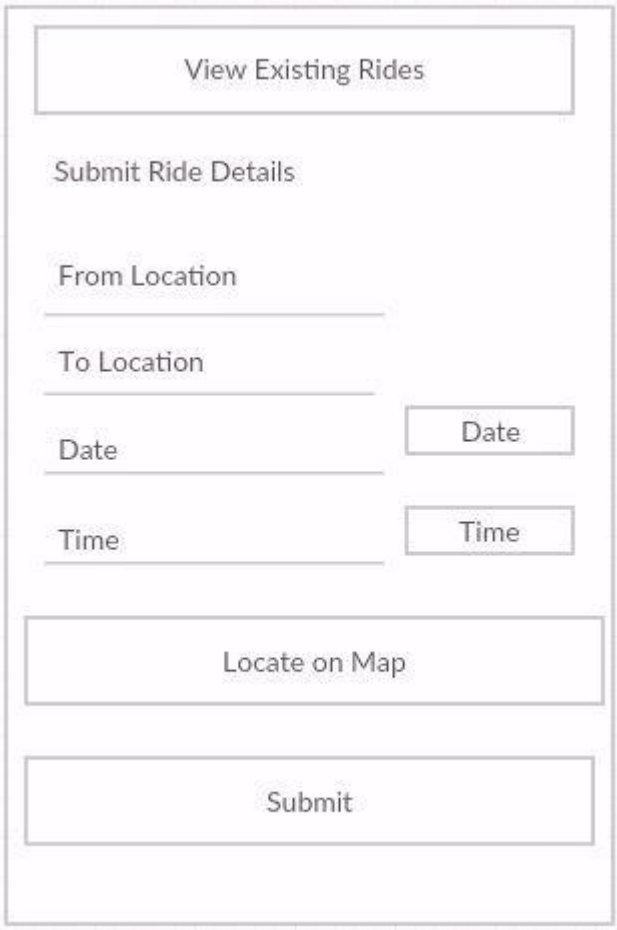


4. Settings page is used by the user to update details like password, phone number and other details

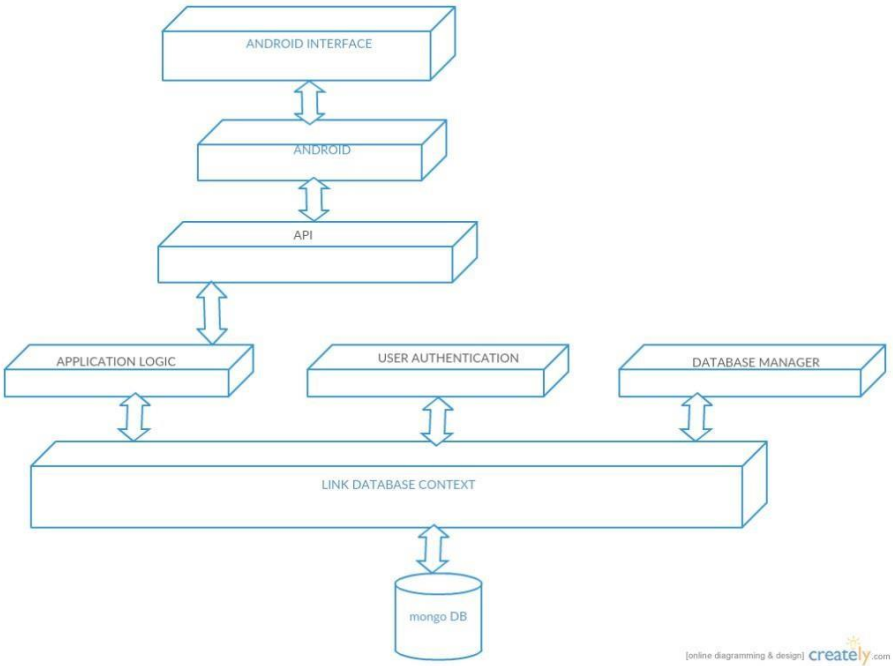


5. Take Ride and View existing Rides page would display the past rides of that user which allows the user to rate the rider.

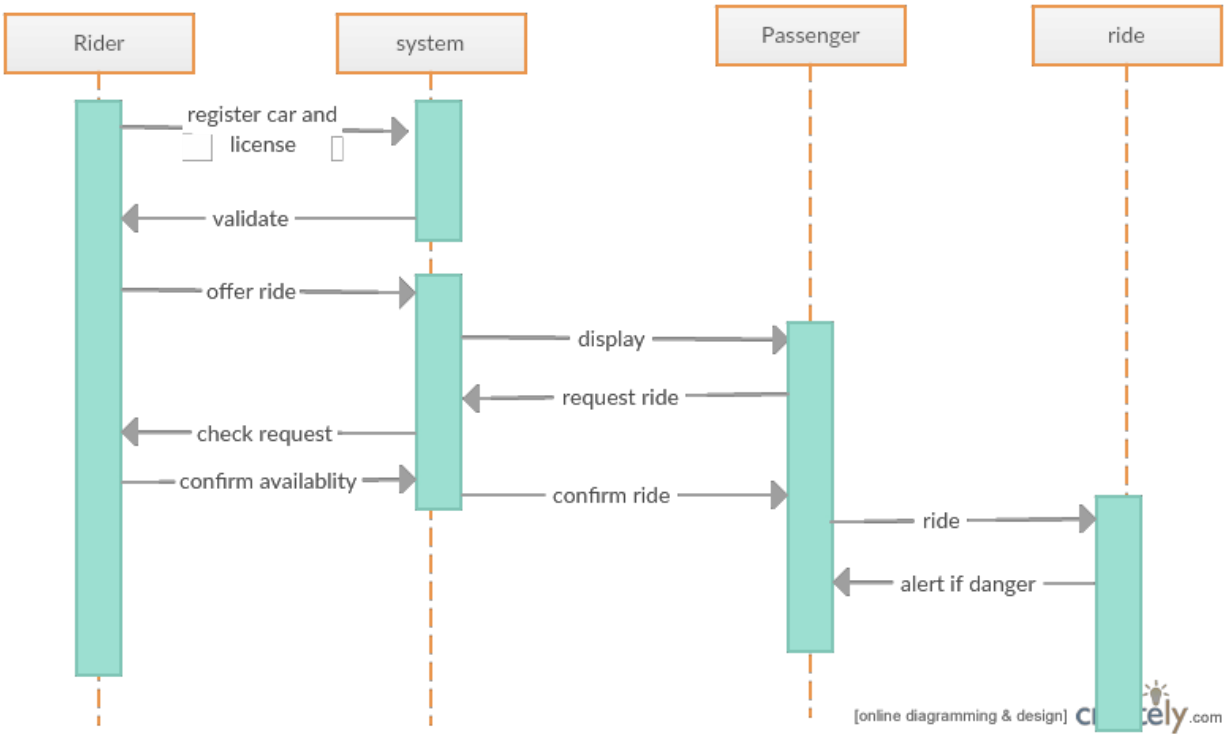
6. Request ride page is used for the user to search/ create rides.



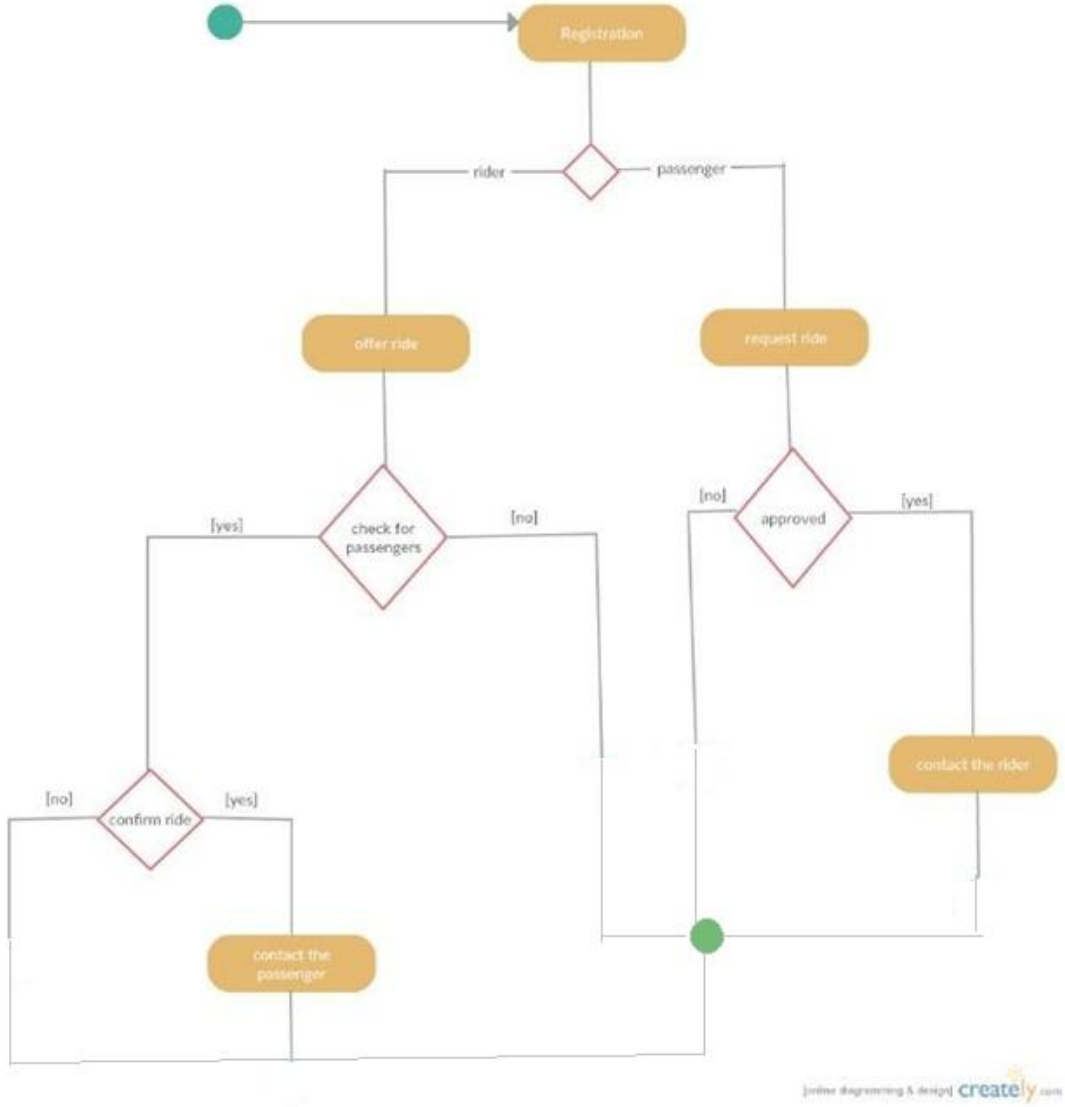
ARCHITECTURE DIAGRAM:



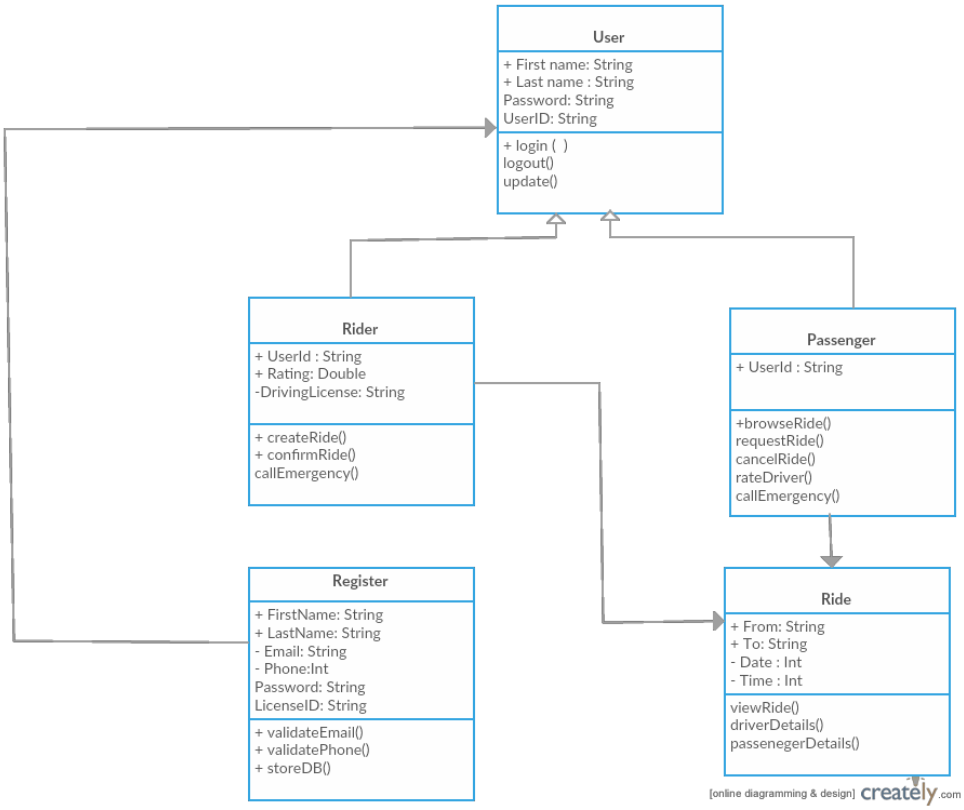
SEQUENCE DIAGRAM:



ACTIVITY DIAGRAM:



CLASS DIAGRAM:



USER STORIES:

1. US-1 (Login details of the Application)

User-Story Description: The user should be able to view the signing page enabling him to either login if an existing user or registering if new user. User can login with Gmail or Facebook.

Actor: Users

Requirement Specification: The screen should consists of two fields accepting the Email and password and two buttons login and New-user. Gmail and Facebook login buttons as well.

- Should have an input text field to accept Email
- Should have an input password field to accept password
- Button named Login, when clicked by the user he should be directed to US-2
- Button named New-user, when clicked by the user he should be directed to US-3
- The details of the user are authenticated, if he/she are existing users.
- Page should consists of Gmail and Facebook buttons enabling the user to login with existing accounts.

Assumptions: User should be able to perform the login and registration successfully.

2. US-2 (Home screen of the users)

User-Story Description: The user can see his active rides if he has currently booked any and he can further choose if he wants to give a ride or take a ride. He can logout at any time and can send a message on the hit of a button during emergency. User can also manage his settings.

Actor: Users

Requirement Specification: The screen should consist of fields displaying the users past rides, it should further have buttons to request rides, offer rides, past rides, manage settings, logout and an emergency button.

- Should have a side menu to display different buttons.
- Button named Manage Settings when clicked by the user should be directed to US-4.
- Button named Logout to come out of the application.
- Button named Request Ride when clicked by the user should be directed to US-5.
- Button named Offer Ride when clicked by the user should be directed to US-6.
- Button named Past History when clicked by the user should be directed to US-7.
- Button named Emergency to send a message to contacts during emergency.

Assumptions: User should be able to perform the login and use functionalities successfully.

3. US-3 (Registering the new-users)

User-story Description: Enabling the user to register into the application by taking the basic details.

Actors: Users

Requirement Specifications: As a New-user, he should be able to register with the application by filling the details of the user.

- Screen should consist of an input text field to accept the first name of the user
- Should consist of an input text field to accept the last name of the user.
- Should consists of an email field to accept the email.
- Should consist if an input text filed to accept username.
- Should consist of an input filed to accept mobile number.
- Should have a set password field enabling the user to set password
- Should have a confirm password field to verify the password.
- Should have a button to upload other proofs.
- Should have button to upload the user's driver's license.
- Should have a button to upload the users profile photo.
- Button to submit the details to database.

The following fields should be mandatory:

First name, email, Password, passport, mobile number, username

Assumptions: New-users should be able to create an account in the application.

4. US-4 (Update users profile)

User-Story Description: The user should be able to change his password, mobile number, email and he can also add the contact details of his family to send them a message during emergency.

Actor: Users

Requirement Specification: The screen should consists of fields accepting the Email, password, mobile number, other info and two buttons login and New-user.

- Should have an input text field to enter new password.
- Should have an input field to enter new phone number.
- Should have an input field to enter new email.
- Should have an input filed to update other info.
- Should have 3 input fields to enter name and mobile number.
- Should have a button save to successfully update the details to the database.
- Should have a button home to enable the user to move to home page.

Assumptions: User should be able to perform the update and save details successfully.

5. US-5 (Request ride)

User-story Description: The user should be able to request ride between two places.

Actor: Users

Requirement Specifications: The screen should consist of the following fields and buttons

- Should have an input text field to enter the starting point.
- Should have an input filed to enter the destination.
- Should have an input field to enter the time.
- Should an input filed to enter the date of journey.
- Should have a button Request Ride to request a ride to the rider.
- Should have a link which on click should display the existing rides available between the two locations.
- Should have a button Home to redirect to the home page.

Assumptions: User should be able to request the ride successfully.

6. US-7 (View existing rides)

User-story Description: The user should be able to view his past rides and give ratings to rides.

Actor: Users

Requirement Specifications: The screen should consist of the following fields and buttons

- Should have an input text box to display the users past rides.
- Should have a field to rate the driver of his previous ride.
- Should have a Save button to save the updated rating.
- Should have a home to populate user to home page.

Assumptions: User should be able to view all his rides.

7. US-6 (Offer Ride)

User-story Description: The user should be able to offer ride details and view the active rides available for him

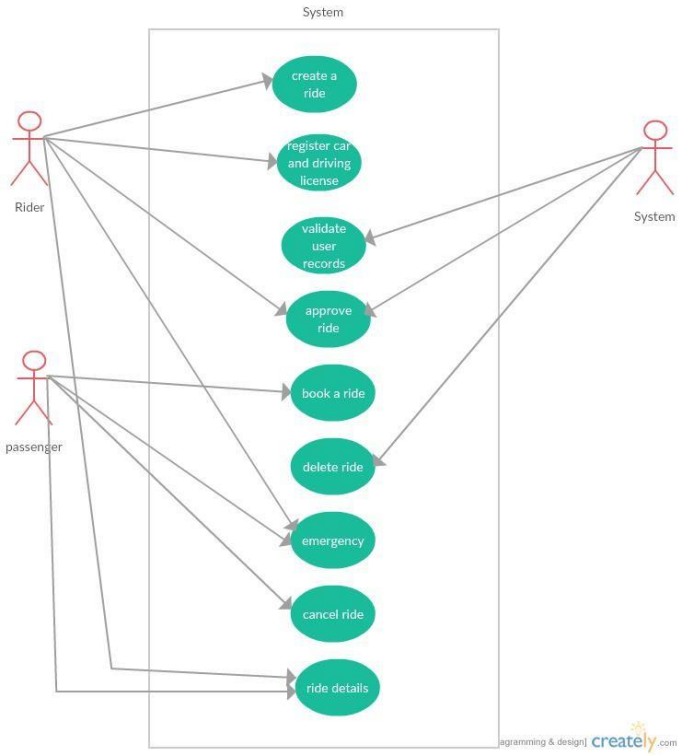
Actors: Users

Requirement Specifications: The screen should consists of the following inputs.

- Should have an input text field to enter the starting point.
- Should have an input filed to enter the destination.
- Should have an input field to enter the time.
- Should an input filed to enter the date of journey.
- Should have a button Offer Ride to request a ride to the rider.
- Should have a link which on click should display the existing rides available between the two locations.
- Should have a button Home to redirect to the home page.

Assumptions: User should be able to request the ride successfully.

USE CASE DIAGRAM:



SERVICE DESCRIPTION:

1. Google API is used to locate the present location of the user and also enabling the New-users to login with Gmail.
2. Used Mongo DB to store the user details and to retrieve the user details when needed.

TESTING DOCUMENTS:

1. Login details of the Application

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to login/New-user should be able to register	The user would be able to enter the E-mail and password and click login to enter to application	User should be able to login	User is logged-in.
2.	User should be able to login/New-user should be able to register	New-User should click the Sing-up button to register, to navigate to registration page	New-users are directed to register page.	New-Users are able to view the registration page.
3.	User should be able to login/New-user should be able to register	User should enter E-Mail and Password and are validated for the users.	User details are authenticated.	User is able to login only when accurate details are submitted

4.	User should be able to login/New-user should be able to register	User is entered wrong details an error would pop up.	User credential are to be validated and if wrong a pop-up would be display	User would be able to view a message if incorrect details are entered.
5.	User should be able to login/New-user should be able to register	User would be able to sign-in with Gmail or Facebook.	User should be successfully logged in.	New-users are able to login.
6.	User should be able to login/New-user should be able to register	User can read about the application following that link	User can view the information about the application.	User is able to view the application related data.

2. Registration screen of the Users

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	New-User should be able to register by giving the basic details.	New-Users should fill all the fields and click submit.	New-users should be successfully register.	New-Users are registered.
2.	New-User should be able to register by giving the basic details.	New-User should enter all the mandatory fields. First name, email, password, passport	New-users are registered once the user enters the mandatory fields	New-Users are able register.
3.	New-User should be able to register by giving the basic details.	New-Users would be notified with validation regarding the mandatory fields.	New-Users are able to view an error if fields are left blank.	New-users are notified with error.
4.	New-User should be able to register by giving the basic details.	On submit the details entered are saved to the Data Base.	New-User details are save to the data base once they click the submit button.	Details of new users are save to data base.

3. Ride

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	Registered User should be able to either request a ride or give in ride details.	Users can fill in the ride details and submit their details.	Users should be able to fill the ride details to provide a ride	Users can successfully provide their ride details.
2.	Registered User should be able to either request a ride or give in ride details.	Users can click request ride to view available rides	Users can view available rides	Users are able to view available rides
3.	Registered User should be able to either request a	User can click offer to create a ride	Users should be able to create rides	Users are able to create the rides

	ride or give in ride details.			
4.	Registered User should be able to either request a ride or give in ride details.	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully.

4. Home screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to choose different services	Users should click request ride to redirect to ride page.	Users should be able to redirect to the ride page	Users are successfully redirected.
2.	User should be able to choose different services	Users should click offer ride to redirect to the ride page	Users should be able to redirect to the ride page	Users are successfully redirected.
3.	User should be able to choose different services	Users should click past rides to redirect to the previous rides page	Users should be redirected to the past rides page	Users are successfully redirected to the page.
4.	User should be able to choose different services	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully
5.	User should be able to choose different services	Users can click manage settings button to redirect to manage settings page	Users should be able to redirect to the page	User are successfully redirected

5. Manage settings screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to update his details	Users should enter any field that he wants to be update	Users should be able to update the information	Users have successfully updated the information.
2.	User should be able to update his details	Users should be able to redirect to home page	Users should be able to redirect to the ride page	Users are successfully redirected.
3.	User should be able to update his details	User can enter the emergency contacts and click save	Users should be able to redirect to home page	Users are successfully redirected to home page

6 .Past rides

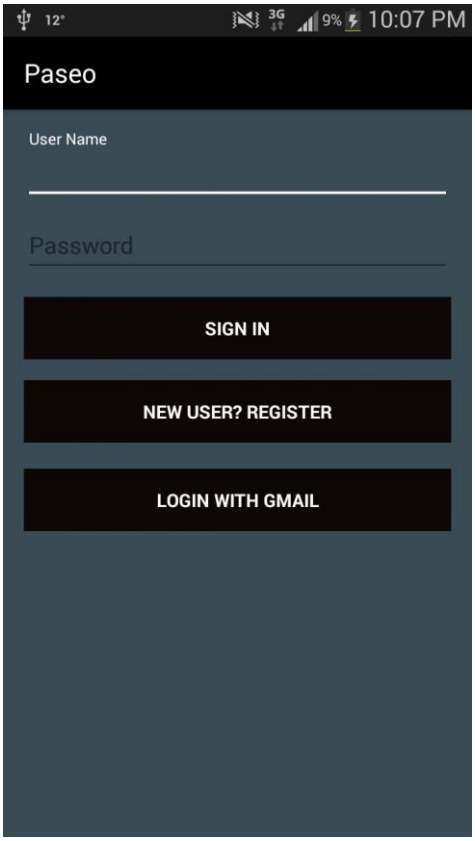
S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to rate their previous rides	Users should rate their previous ride and click save.	Users should be able to redirect to the home page	Users are successfully redirected.

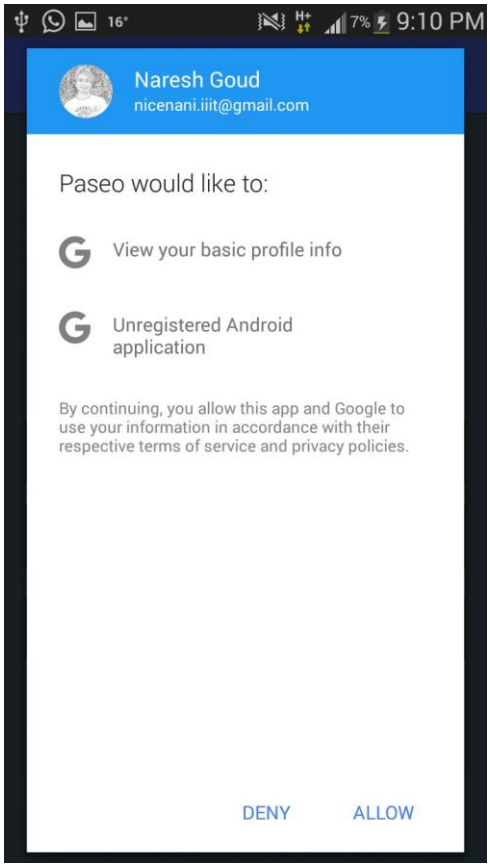
IMPLEMENTATION:

- Mobile client Implementation: Implementation in mobile is feasible and is easy to use as most of the present world runs on smart phones. We have implemented the project using Android.
- Server Implementations: Used Mongo database to store and retrieve the details of the user about his ride and profile info.

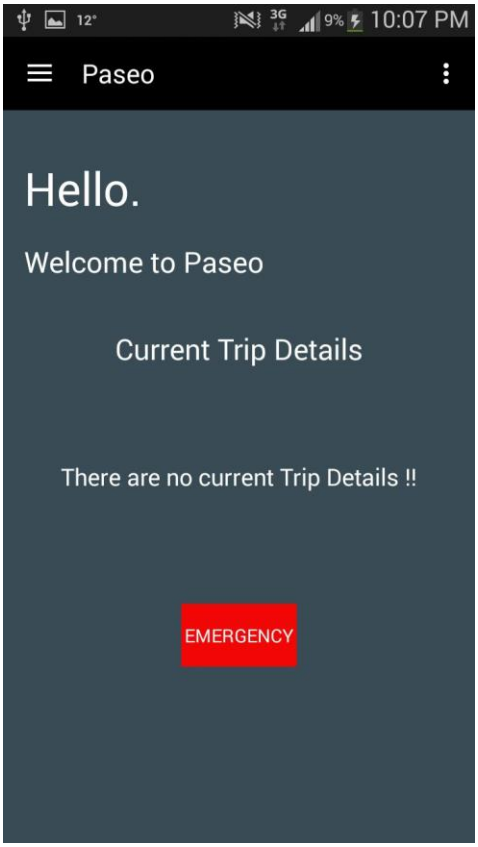
DEPLOYMENT:

1. Login Page:

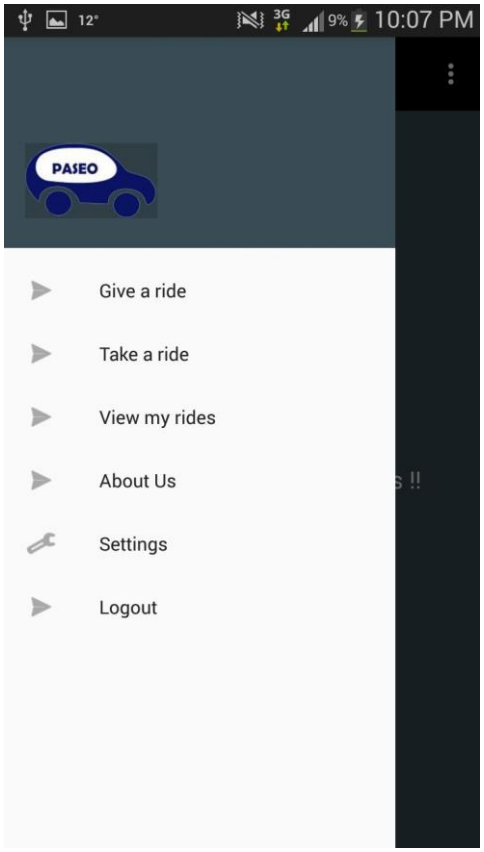



16°
7%
9:10 PM
Naresh Goud
nicenani.iiit@gmail.com
Paseo would like to:
View your basic profile info
Unregistered Android application
By continuing, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies.
DENY
ALLOW

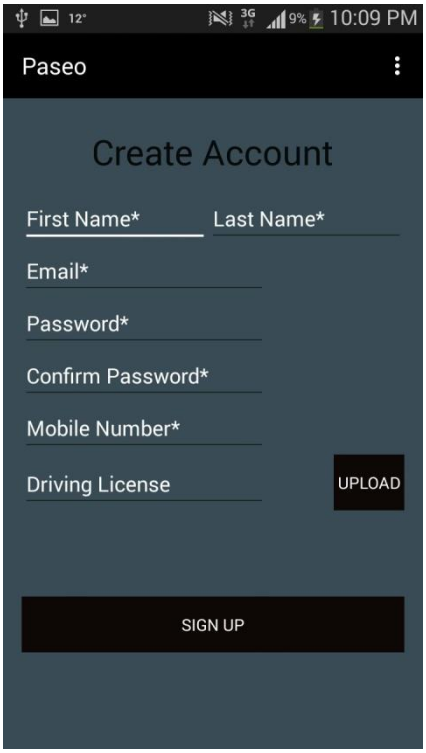
2. **Home Page:**



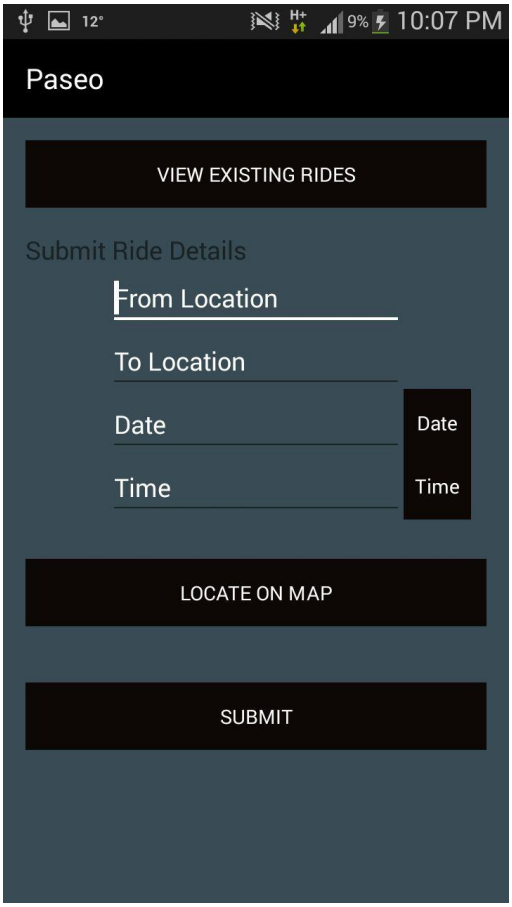
Menu Screen:

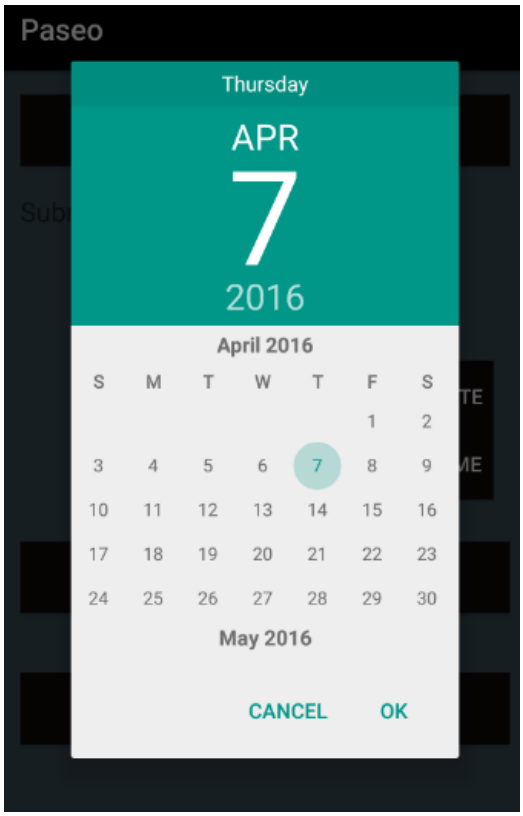


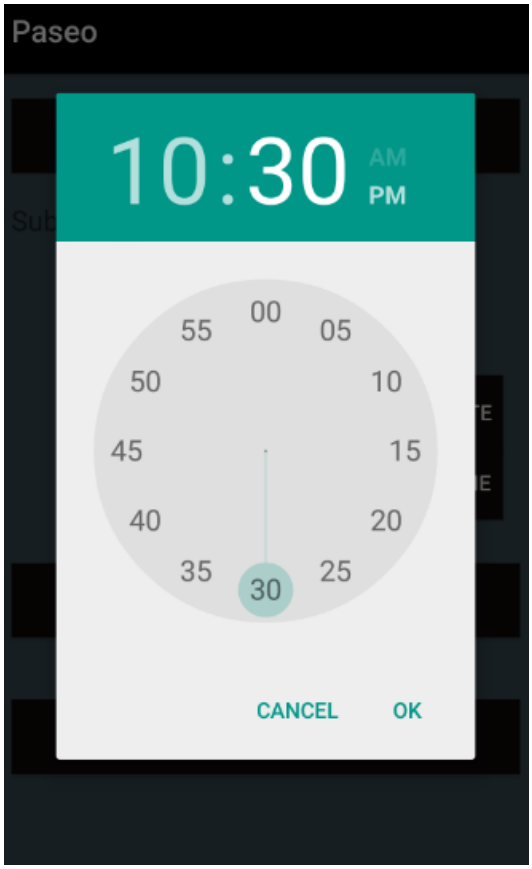
3. **Registration Page:**



4. **Ride Request:**

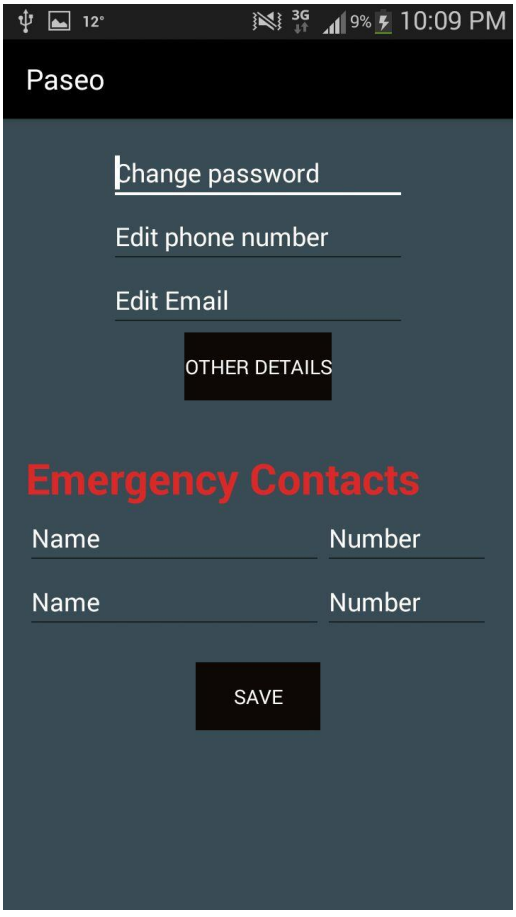



Paseo
Thursday
APR
7
2016
April 2016
S M T W T F S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
May 2016
CANCEL
OK

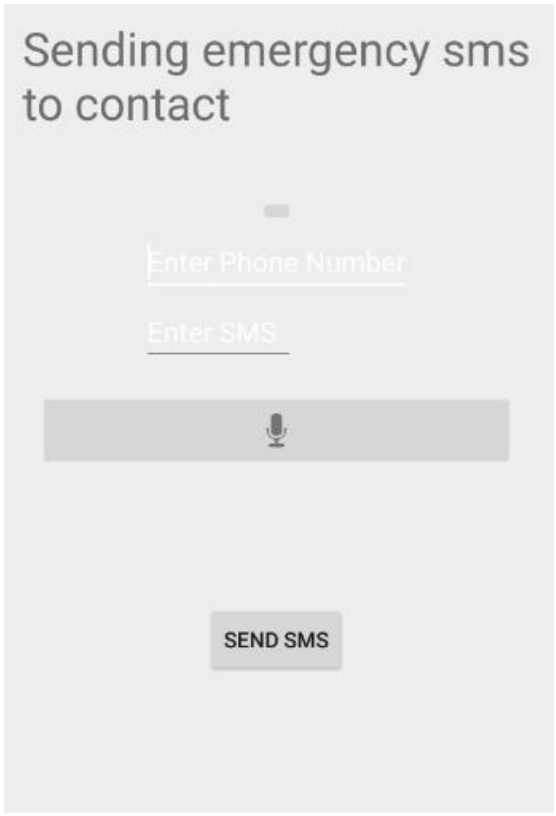

Paseo
10:30
AM
PM
00
05
10
15
20
25
30
35
40
45
50
55
CANCEL
OK

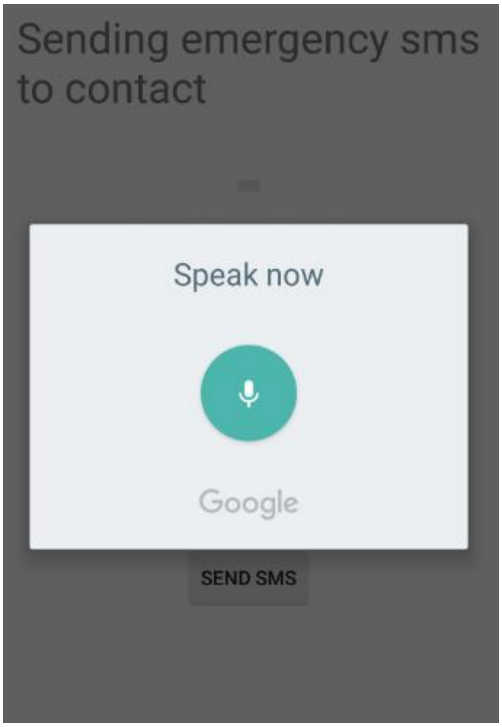
5. **Offer Ride:**

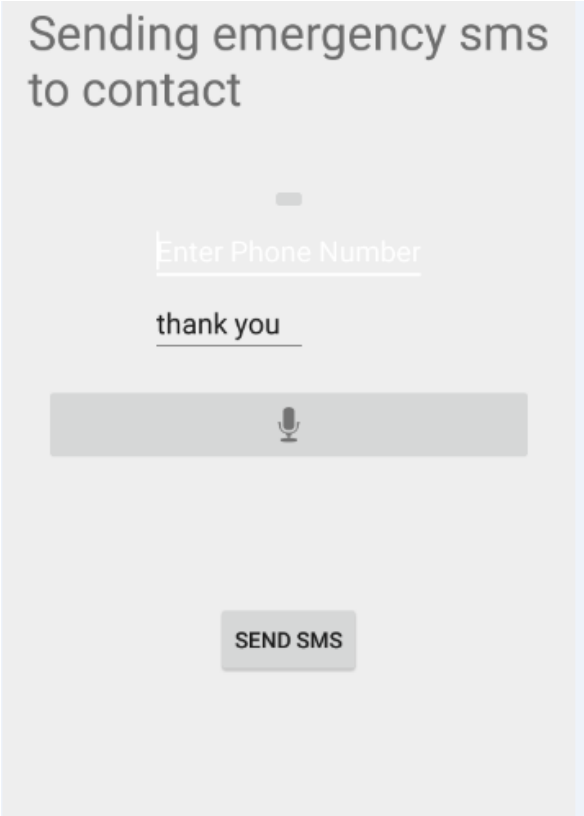
6. **Settings**



7. **Security features:**







PROJECT MANAGEMENT:

- **Project Timeline: 7th** April 2016.
- **Members:** Amulya1, VinuthaMuthyala, Naresh, Megha Sai.
- **Task Responsibilities:** Issues are created in github for each members and every member have collectively worked to complete them in given time. Have shared the work equally among us.

Work

Completed:

Description:

- **Stories:** UI is improved, security alert notifications, emergency contacts, voice to text and the feedback rating is implemented.
- **Service Design:** The Google service is implemented to notify the user with the current location and Mongo database service to save and retrieve the information of the user and ride, they are implemented to enhance the functionality of the user.
- **Service Implementation:** Google API and Mongo database and are being worked successfully.

Responsibilities:

We all have worked together to provide an effective and clear implementation of the project.

Amulya Pindi: Sign-in with Gmail, sending SMS and notifying the user which are a part of emergency, giving feedback to the riders, UI of the settings page, mandatory fields in application and documentation.

Vinutha Muthyala: UI part of the home page, Voice to text conversion as a part of emergency, Functionality of retrieving the data from Mongo database, Google Map services and driver's document validation.

Megha Sai: Architecture/use case diagrams, UI, design of request screens, wireframes and manage setting features.

Naresh: To store the data to Mongo database, UI, Activity diagram and manage settings features.

Time Taken and Contributions: All have contributed 5hrs each almost every alternate day and almost 10hrs completely the last three days.

Work to be completed:

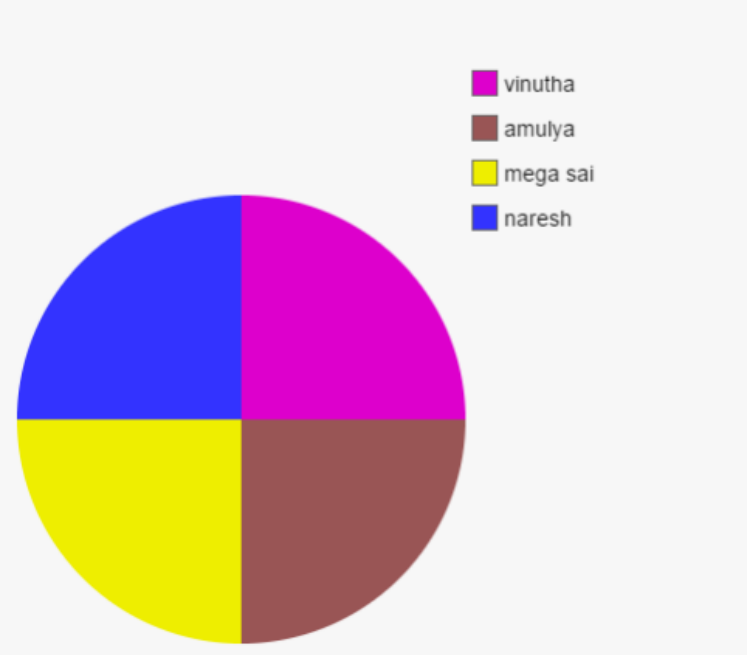
Description:

Payment features, location alarm implementation, directions in google maps, enhancements in security features.

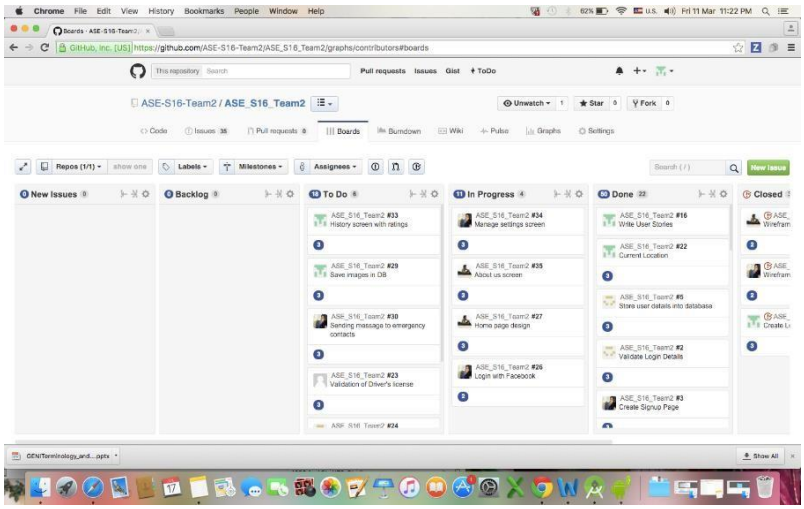
Issues:

Experienced few issues in implementing the notifications in security, route implementation in google maps and about the payment feature implementation.

Burndown Graphs:



DASH BOARD:



BIBLIOGRAPHY:

1. https://en.wikipedia.org/wiki/Carpool
2. https://en.wikipedia.org/wiki/Real-time_ridesharing
3. http://www.carpoolworld.com/
4. http://www.ijarcsse.com/docs/papers/Volume_3/4_April2013/V3I3-0385.pdf
5. http://www.ijsce.org/attachments/File/v5i2/B2612055215.pdf